



Universidade Federal De Santa Catarina  
Departamento de informática e estatística

Farkle

Especificação de requisitos de software

Identificador: 2103201802

Versão: 0.6

Data: 12/04/2018

### Histórico do documento:

Versão:	Autore(s):	Data:	Ação:
0.2	David Grunheidt Luiz Valdiero	28/03/2018	-Início da inclusão dos requisitos
0.4	David Grunheidt Luiz Valdiero	06/04/2018	-Introdução da seção “Organização da especificação”. -Explicitação quanto às regras comuns na seção de documentos externos -Referenciação das regras de negócio pelos requisitos funcionais
0.6	David Grunheidt Luiz Valdiero	12/04/2018	-Remoção da seção “Organização da especificação”. -Alteração nos requisitos funcionais quanto as referências as regras de negócio.

## • ÍNDICE:

1. Introdução ao software
  - 1.1. Objetivo do desenvolvimento
  - 1.2. Definições de abreviações e termos
  - 1.3. Documentos externos importantes
  - 1.4. Localização desta especificação na base de dados da empresa
2. Visão geral do sistema
  - 2.1. Arquitetura da aplicação
  - 2.2. Premissas de desenvolvimento
3. Requisitos da aplicação
  - 3.1. Requisitos funcionais [RF]
  - 3.2. Requisitos não funcionais [RNF]
  - 3.3. Regras de negócio [RN]
4. Esboço da interface gráfica

## I. INTRODUÇÃO:

### 1.1- Objetivo do desenvolvimento:

Esta especificação tem como objetivos principais esclarecer todas as possíveis funcionalidades e aplicações do software “Farkle”, jogo de dados em que, resumidamente, o objetivo é jogar dados e obter com estes pontuações baseadas em determinadas regras até chegar em um certo número de pontuação, assim como descrever, de forma clara e precisa, como se dará o funcionamento deste a partir de seus elementos, tanto internos quanto externos, e suas interações.

### 1.2- Definições de abreviações e termos:

- **Dado:** Neste software será simulado uma peça importante dos jogos de tabuleiro, o “dado” em forma de um cubo (Poliedro possuindo 6 lados (hexaedro), onde cada um destes lados é um quadrado) no qual, cada lado desse cubo representa um número e estes números têm relação direta com as diferentes combinações de pontuações possíveis de se obter.
- **“Farkled”:** Se relaciona com o fato de um jogador ter perdido todos os pontos feitos em um determinado round.
- **“Rounds”:** São cada um dos períodos em que se divide um jogo, onde todos jogadores tem sua vez de jogar enquanto os outros esperam.
- **“Roll”:** Ação de “jogar” os dados a fim de se obter um dos “n” números presentes em cada um deles.
- **“Set Aside”:** É possível também selecionar ou desselecionar um determinado dado a fim de adicioná-lo aos dados já selecionados, os quais poderão ou não resultar em dados “setAside” e consequentemente, uma pontuação.
- **“Bank”:** este termo se refere a outra ação importante, a qual o jogador termina o seu “round” e acumula a pontuação deste round a sua pontuação total.
- **“Hot Dice” ou “Free Roll”:** ocorre quando o jogador pontua todos os dados.
- **“Pontuação total” ou “Grand Total”:** Pontuação que acumula as pontuações de todos “rounds”.

### 1.3- Documentos externos importantes:

Vale salientar importância especial quanto a leitura de pelo menos um dos documentos externos referenciados nesta seção, já que estes contêm as regras comuns do jogo a ser simulado, as quais só serão descritas aqui. As demais regras criadas em especial para este software serão descritas na seção de regras de negócio.

[1]. Wikipedia. Farkle. Abril, 2010. Disponível em: <https://en.wikipedia.org/wiki/Farkle>.

[2]. Smart Box Design LLC. Farkle Rules. Disponível em: <http://www.smartboxdesign.com/farklerules.html>

#### **1.4- Localização desta especificação na base de dados da empresa:**

Para verificar a localização desta especificação no nosso banco de dado e, visualizar especificações anteriores ou de versões mais recentes, basta acessar o link a seguir: <https://goo.gl/nIM29f>. Nesta pasta estão todas as especificações de todas as versões do software “Farkle”, inclusive esta versão, v0.4.

### **II. VISÃO GERAL DO SISTEMA:**

#### **2.1- Arquitetura do programa:**

Software com lógica de implementação voltada para orientação a objetos e um sistema distribuído do tipo cliente-servidor.

#### **2.2- Premissas de desenvolvimento:**

- 2.2.1.** A implementação do software deve ser feito de modo que possa ser executado em qualquer máquina que possua instalado a máquina virtual java, de modo que deve também ser implementado em linguagem de programação Java.
- 2.2.2.** O software precisa apresentar uma interface gráfica de caráter simples, simulando um cenário bidimensional (2 dimensões).
- 2.2.3.** Interações entre diferentes usuários através da rede devem ser permitidas pelo programa, o qual deverá utilizar-se de um sistema distribuído cliente-servidor através da ferramenta NetGamesNRT.

### **III. REQUISITOS DA APLICAÇÃO:**

#### **3.1- Requisitos funcionais [RF]:**

**[RF 1] - Conectar:** O programa deve apresentar opção de “Conectar” para que o jogador possa se conectar em um servidor, será exigido o nome do jogador e o servidor do jogo. Quando conectado o jogador poderá ser alocado numa partida.

**[RF 2] - Desconectar:** O programa deve possuir a opção de “Desconectar”, para se desconectar do servidor. O jogador pode desconectar-se somente se estiver conectado segundo o **(RF 1)**. Esta

funcionalidade é acessível através de 2 diferentes contextos. Se o jogador estiver conectado mas fora de um contexto de partida iniciada, ele se desconecta a partir da opção “Sair do jogo” (RF 13). Se o jogador estiver conectado e em um contexto de partida iniciada, então este se desconecta sobre as regras (RN 11) a partir da opção “Abandonar Jogo” (RF 12).

[RF 3] - **Criar partida:** O programa deve permitir criação de partidas. Esta criação caracteriza-se como uma tentativa, por parte de um determinado jogador, de iniciar um novo jogo. Para solicitar que o servidor crie uma partida, é necessário, primeiramente, estar conectado, e, após isto, indicar o número de jogadores máximo desejado para esta partida, respeitando a (RN 1).

[RF 4] - **Iniciar Partida:** O programa deve iniciar a partida após o número máximo de jogadores conectados ao servidor definidos previamente estiver completo, ou seja, quando o que foi definido através do (RF 3) baseando nas regras (RN 1) seja atingido, e, após ser realizado a votação para escolha do nível referenciado no (RF 5) baseando nas regras sobre os níveis (RN 2). Além disso, num último momento antes da instanciamento das classes devidas para caracterização de um contexto de “partida iniciada”, o programa deve definir a ordem dos jogadores como referenciado na regra de negócio (RN 3).

[RF 5] - **Votação entre jogadores para definição de nível:** Cada jogador votará em um nível baseados na (RN 2). O nível escolhido será igual ao inteiro resultante do cálculo da média dos votos, onde cada voto será representado na forma de um inteiro de 1 a 3 (sendo 1 o nível fácil, 2 o médio e 3 o difícil).

[RF 6] - **Placar:** O programa deve mostrar a pontuação “grandTotal” de todos os jogadores, e a pontuação do “round” e do “roll” do jogador da vez, conforme referenciada na (RN 7).

[RF 7] - **Comunicação do andamento da partida:** O programa deve manter todos jogadores atualizados quanto ao andamento da partida, ou seja, todas as regras de negócio descritas nas (RN 4), (RN 5) e (RN 6), onde, nestas três primeiras regras será possível visualizar os dados “definidos” após um “roll” e os “selecionados”, (RN 7), (RN 8), (RN 9), as quais, o botão de “bank” só deve ser visto pelo player da vez, porém, quanto a o que foi “conquistado” por este no final de seu round deve ser visto por todos, (RN 10), (RN 11) e, por fim, (RN 12).

[RF 8] - **Fazer um “Bank”:** O programa deve permitir coletar os pontos acumulados no “round” a partir do botão “Bank”, fazendo as devidas verificações de acordo com a (RN 8). Após a coleta, deve ser verificado se o jogador venceu a partida, passando a vez ao próximo jogador caso não, ou informando a todos os jogadores, como descrito na (RN 12), que ele foi o vencedor, caso sim. Restrição: ser o jogador da vez.

[RF 9] - **Selecionar dados:** O programa deve permitir selecionar algum conjunto de dados dentro os que foram “rolled”, colocando este conjunto com status “selecionado”, como definido na

(RN 5). Para concretizar a pontuação desejada o jogador deve então dar um “roll” ou “bank”, onde será verificado se os dados selecionados são válidos.

[RF 10] - **Fazer um “Roll”:** O programa deve permitir fazer o arremesso de “n” dados livres. Tanto a regra (RN 4) quanto a (RN 6) utilizam-se desta funcionalidade. Restrição: Ser o jogador da vez.

[RF 11] - **Ajuda:** O programa deve permitir consulta a uma tabela de exemplos de pontuação. Esta funcionalidade deve mostrar algumas combinações de dados e as pontuações equivalentes a elas conforme referenciadas nos documentos externos que explicitam as regras comuns.

[RF 12] - **Abandonar jogo:** O programa deve permitir abandono de jogo conforme a (RN 11). O jogador sairá da partida, e automaticamente a partida se encerra para todos os outros jogadores, mostrando o jogador com maior pontuação como o ganhador. Esta opção estará presente quando o jogador estiver numa partida.

[RF 13] - **Sair do jogo:** O programa deve permitir sair do jogo.. Esta opção só estará presente quando o jogador não se encontra conectado ao servidor segundo o (RF 1).

[RF 14] - **Fim de jogo:** O jogo deverá terminar quando houver um vencedor (RN 12) ou quando um jogador desconectar da partida segundo a regra (RN 11).

### 3.2- Requisitos não funcionais [RNF]:

[RNF 1] - **Linguagem de programação:** O código deverá ser feito em linguagem java.

[RNF 2] - **Especificação de projeto:** A especificação deverá ser baseada em UML2.

[RNF 3] - **Tecnologia da interface gráfica:** O pacote ou biblioteca “Swing” do Java deverá ser utilizada na criação da interface gráfica.

[RNF 4] - **Padrão de projeto MVC:** A implementação deve ser feita de modo a separar o modelo do domínio, a apresentação e as ações do usuário em 3 diferentes áreas, modelo, visão e controle. Assim, a organização é mantida em alto nível também na codificação. Simplificando, cada classe deve ser mantida junto com suas equivalentes segundo este padrão de projeto dentro de uma determinada pasta, dentro do projeto.

[RNF 5] - **Padrão de projeto Strategies:** Padrão de projeto que define variações de um algoritmo, onde este está relacionado, no caso do programa desta especificação, com a dificuldade do jogo ou da partida em questão.

[RNF 6] - **Padrão de projeto Factory:** Este padrão é responsável por criar objetos sem que a lógica de criação seja exposta ao cliente. Simplificando, uma interface ou classe abstrata é definida para a criação de um objeto mas suas subclasses que decidirão qual tipo daquele objeto será criado. Complementa com o padrão strategies.

### 3.3- Regras de negócio [RN]:

[RN 1] - **Número de jogadores:** Todas partidas devem ter de dois a quatro jogadores.

[RN 2] - **Nível de dificuldade:** O jogo terá 3 níveis de dificuldade diferentes, fácil, médio e difícil, onde cada um deles tem relação direta com o número de dados da partida, os métodos para verificação de pontuação e os métodos para dar um “roll” segundo o número de dados. As regras definidas para cada estão explicadas a seguir.

- Nível Fácil: Cada jogador possui 6 dados e é penalizado em 300 pontos no “three farkled”. Para dar bank é necessário ter 200 pontos mínimos no round. Ganha o primeiro que chegar a 5000 pontos.
- Nível médio: Cada jogador possui 5 dados e é penalizado em 400 pontos no “three farkled”. Para dar bank é necessário ter 250 pontos mínimos no round. Ganha o primeiro que chegar a 6000 pontos.
- Nível difícil: Cada jogador possui 5 dados, é penalizado em 500 pontos “three farkled”. Para dar bank é necessário ter 300 pontos mínimos no round. Ganha o primeiro que chegar a 7000 pontos.

[RN 3] - **Vez dos jogadores:** O programa deve ser capaz de decidir a ordem dos jogadores, e, baseando-se nesta definição, referenciar tanto o jogador da vez quanto os jogadores que não a possuem.

[RN 4] - **Primeira jogada num “round”:** O jogador começa sua vez no “round” podendo fazer um “roll”, os próximos só poderão ser feitos se o jogador ter pontuado o último “roll”.

[RN 5] - **Separação dos dados que pontuaram:** Ao jogar os dados (“roll”), se o jogador tiver obtido uma combinação de “n” dados que gere alguma pontuação (de acordo com as regras comuns de pontuação vistas nos documentos externos), ele poderá selecionar 1 a “n” dados a fim de obter determinada pontuação. A verificação do conjunto dos dados selecionados para determinar se é uma combinação válida para pontuar é feita quando o jogador tenta realizar “bank” ou “roll”. Quando os dados são verificados e dados como válidos, são marcados como “setAside” e não podem mais serem desselecionados.

[RN 6] - **“Rolls” subsequentes no mesmo “round”:** O jogador pode fazer quantos “rolls” puder em um mesmo “round”, desde que, pontue todos dados selecionados do “roll” anterior, tendo selecionado pelo menos um dado. Quando o jogador fizer um “Hot dice” ele terá novamente todos dados iniciais e poderá continuar fazendo novos “rolls”.

- [RN 7] - **Pontuação:** A pontuação é organizada da seguinte forma, pontuação de “roll”, pontuação de “round” e pontuação “grand total”. A pontuação “grand total” acumula as pontuações do “round” adquiridas no final de cada “round”, e a pontuação do “round” acumula as pontuações dos “rolls” durante o “round”.
- [RN 8] - **Finalização de um round com pontuação:** O jogador só poderá fazer um “banking” se tiver pontuado todos dados selecionados do “roll” anterior, tendo selecionado pelo menos um dado, e ter uma pontuação mínima no round de acordo com o nível. Ao realizar um “banking” o jogador passa a vez para o próximo jogador.
- [RN 9] - **Finalização de um round sem pontuação:** Ocorre um “farkled” quando o jogador não consegue pontuar após realizar um “roll”, ele perde os pontos acumulados no “round” e passa a vez para o próximo jogador.
- [RN 10] - **“Three Farkled”:** O “three farkled” penaliza o jogador da vez em pontos (que variam de acordo com o nível de dificuldade da partida) a cada três “farkleds” em “rounds” seguidos.
- [RN 11] - **Abandono de partida:** O jogador pode abandonar a partida em qualquer momento.
- [RN 12] - **Vencedor:** O vencedor será o jogador que alcançar dez mil pontos.

#### IV. ESBOÇO DA INTERFACE GRÁFICA:



# Farkle

Conectar

Sair

Ajuda

# Farkle

Criar Novo Jogo

Desconectar

Ajuda

# Farkle

Criar Novo Jogo  
Número de Jogadores

...

Sair

Ajuda

Menu

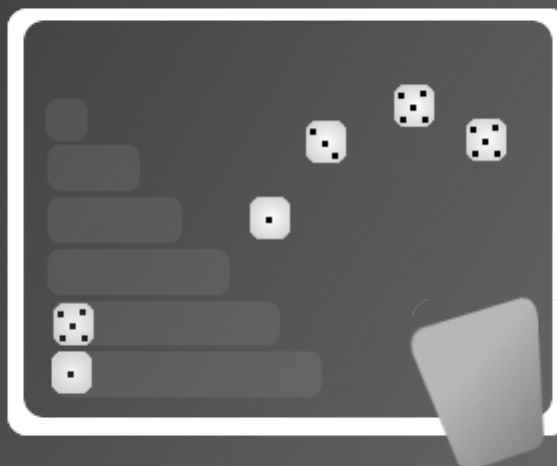
3 - João

200

200

Bank

# Farkle



Jogador da vez: Maria

150

1- Pedro

800

2 -Maria

650